

Wire Cell / LArSoft Integration

Brett Viren

Physics Department



BROOKHAVEN
NATIONAL LABORATORY

The logo for Brookhaven National Laboratory features the word "BROOKHAVEN" in a bold, black, sans-serif font. Below it, the words "NATIONAL LABORATORY" are written in a smaller, black, sans-serif font. A stylized, grey, curved line with a red dot at its end sweeps across the text from the bottom left towards the top right.

Wire Cell Summit 7-9 Dec 2015

Outline

Overlap and Synergy

The Integration Challenge

Some Integration Methods

FNAL Software Infrastructure

Summary

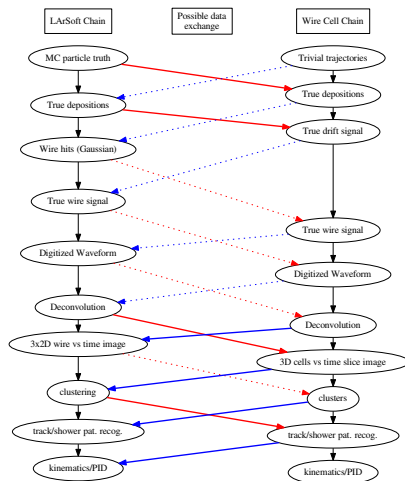
LArSoft and Wire Cell Characteristics Compared

	LArSoft/Art	Wire Cell
arch	framework	toolkit
exec	serial	parallel DFP (or DIY)
sim	Geant4 + the 5 "D"s	just the 5 "D"s ¹
reco	mostly 2D then merge	3D image + pat.rec.
algs	alt/competing algs	one alg/step (no alt)
chain	full MC + data	drift sim + data
perf	fast and good	very slow but better
build	fnal.gov	portable
support	SCD	shoestring
maturity	adult	teenager

¹Deposit, Drift, Diffuse, Detector response, Digitize.

Opportunities for Data Exchange

- In principle, can exchange data between many steps in a fine-grained manner.
- Not all exchanges are profitable, eg, WC trajectories are trivial, LS $3 \times 2D$ imaging may not work with WC clustering.
- Solid red/blue lines show likely fruitful exchange.
- Every line implies real effort:
 - Data model matching.
 - Converters.
 - Exchange channel.



Overlap and Synergy

The Integration Challenge

Some Integration Methods

FNAL Software Infrastructure

Summary

Integration Challenges

The good:

- A file exchange method already in use (celltree).
- Wire Cell (toolkit) is explicitly designed for easy integration at the code level (compiling/linking)

Main challenges:

- Memory** each of LS and WC alone require $\sim 2\text{GB}$ (individual memory usage is largely **not** redundant)
- Fermigrid limits to 2GB/job by default, other clusters similar.
 - Massively parallel systems even more limited in RAM/core (thus shared-memory, multi-threaded WC developed).

Portability ROOT broken on HPC (at least on ANL Mira), general LArSoft/Art portability issues are known.

Expertise need to understand different code bases, designs, data and execution models.

Some Integration Considerations

Challenges aside, we need *some* form of LS/WC integration.

My most important considerations:

- We must not impede ongoing development of LS nor WC.
 - Each are separate projects with own development cultures.
- Confront the realities of hardware architecture limitations particularly those needed for production processing.
 - These are not just LS/WC or even WC-only challenges.
- Facilitate comparison studies between all viable LArTPC reco methods.
 - Integration for studies is likely very different than integration for production processing.

Overlap and Synergy

The Integration Challenge

Some Integration Methods

FNAL Software Infrastructure

Summary

Exchange Files

- Run LArSoft and Wire Cell as separate executables.
- Pass data between the two via files in batch/serial manner.
- Incrementally adds to file bookkeeping overhead.
- Care in avoiding I/O bottlenecks, esp. in prod. proc.
- Effort to code converters, define file formats.
- Already have a good start with “**celltree**” files.
 - Can hold: MC particle truth, energy dep. digitized waveforms, WC image (space/charge points) and....
 - Supporting modules in both LS and WC (prototype and toolkit)

Notes:

- **celltree** file do not depend on neither WC nor LS code. They hold POD + TClonesArray of TH1F.
- Can someone investigate use of **LArLite** files?

Wire Cell in Single LArSoft Module

- Write a WC “application” which is a LS module/algorithm.
- LS compiled/linked against WC libs, can still make use of WC configuration, plugin, execution managers.
- In principle, threading should be okay:
 - LS thinks WC is just a function call.
 - LS single threaded, gives control to WC.
 - WC threads do not run LS code (nor ROOT).
- Single image may hit memory pressures on Grid/HPC.
- On HPC, hits ROOT support problem and general LArSoft/Art portability problems.
- Maybe okay for doing comparison studies on high-memory, high-core workstations running Sci. Linux.
- Development of such a module requires all of FNAL software infrastructure support. (Not it!)

Variation: Wire Cell in Many LArSoft Modules

A dubious idea:

- A WC DFP can be any size, even one node, or even call to a bare node.
- Can, in principle, distribute WC nodes across many LS modules.
- Won't help with memory pressures and reduces parallelism.
- Puts all intermediates into LS data store, but requires fine-grained data model development and converter coding.

It is a lot of work with no obvious benefit right now, but it is an option.

LArSoft client and Wire Cell Distributed Service

- Write a LS module to make network request to a WC server.
- WC server is distributed on local cores/Cluster/Grid/HPC.
- The request transfers LS data to process.
- The response returns WC results.
- Synergy: need some a WC service to support Bee 2.0.
- For HPC: LS lives on high-memory/high-CPU but otherwise conventional “edge” computer. This pattern is followed by other HPC uses (eg, ATLAS Event Service).

Method Recommendations

My personal opinion of the “right” optima:

	files	module	client/server
reco dev	✓		
HPC dev	✓		✓
HPC prod			✓
Grid prod	✓		✓
compare	✓		
detailed	✓	✓	
Bee 2.0			✓

Notes:

- **Maybe** only detailed comparisons need to run WC as a LS module.
- Production processing memory pressure rules out WC as a LS module.
- I think HPC production ultimately requires client/server architecture.
- The backend service of Bee 2.0 likely to share many common components with what is used to support distributed production processing.

Overlap and Synergy

The Integration Challenge

Some Integration Methods

FNAL Software Infrastructure

Summary

Integration into `fnal.gov`

Wire Cell prototype has been built at `fnal.gov`.

- half a day to locate and understand how to set up FNAL's ROOT6 installation (mostly due to my ignorance and just passive time waiting for the right answer from experts on a mailing list).
- 10 minutes to download Wire Cell source, build and run its tests.
- Went according to instructions. No expert tweaks needed despite Sci. Linux not being the dev platform.

Microboone (Jyoti Joshi) is running Wire Cell prototype built in this manner on Fermi Grid nodes.

UPS'ify

Do we need UPS “products” made from Wire Cell prototype and/or toolkit?

Right now, I think “no”. Why?

- No formal Wire Cell releases yet (prototype nor toolkit).
- Would not be useful to WC developers
- Pure end-users are still almost nonexistent and they can build from source no prob.
- Long established, `fnal.gov`-based DUNE projects seem okay without UPS'ed experiment code (beam MC, Fast MC, mgt/globes).

So, it seems there is no rush.

If anyone wants to do it, go for it, but I think it is still too early to be useful.

Overlap and Synergy

The Integration Challenge

Some Integration Methods

FNAL Software Infrastructure

Summary

Summary

- Many options: lots of potential points and methods for integrating LArSoft and Wire Cell.
- Constraints: Integration strategy needs effort, needs to impede neither project, and is best directed toward most likely uses.
- Files: integration based on exchange files most needed now (and largely in place).
- Service: WC-as-a-service seems best for production running
 - Needed for Bee 2.0 anyways, likely lots of overlap.
 - Need a LS client module.
 - **Any volunteers to work on either?**
- UPS packaging of Wire Cell not expected to be needed right now.